

Application of Software Engineering Fundamentals: A Hands on Experience

Cihan Varol, Coskun Bayrak, and Robert Ludwig

Grupo: Ana Flávia Freiria Rodrigues, Beatriz Flausino Machado, Isabella Cristina da Silveira, Raissa Nunes Peret, Pedro Brassi Luccas, Gustavo Silva Orlando, Aurélio Lacerda Sena Junior.

Sumário



01 Introdução

02 Métodos empíricos e técnicas experimentais

03 Análise estatística

04 Medição

05 Projeto de engenharia

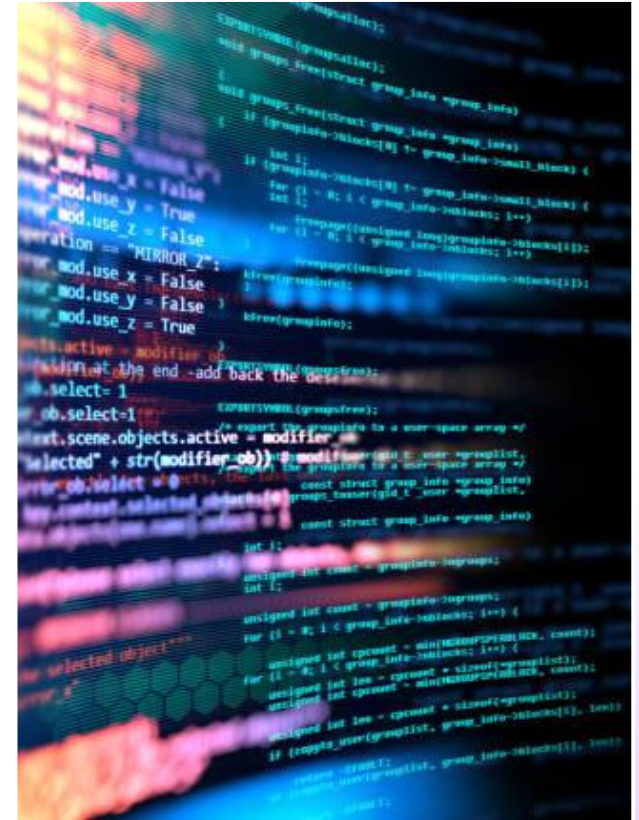
06 Modelagem, simulação e prototipagem

07 Normas

08 Análise e conclusão

01 - Introdução

A engenharia de software exige uma ampla gama de habilidades técnicas para solucionar problemas e projetar soluções. A aplicação adequada desses conhecimentos permite que os engenheiros de software desenvolvam e mantenham softwares de maneira eficiente. Assim, antes de discutir o artigo, serão pontuadas as fundações da engenharia de software, criando uma base sobre o que será abordado na análise do documento.



02 - Métodos empíricos e técnicas experimentais

Experimentos projetados

Testar hipóteses por meio de experimentos controlados para coletar dados e validar teorias.

Estudos observacionais

Observar cuidadosamente os fenômenos do mundo real para identificar padrões e insights.

Coleta de dados

Coletar dados relevantes de forma sistemática e precisa para análise e interpretação.

Análise de dados

Interpretar e analisar dados coletados para obter conclusões significativas.

Estudos retrospectivos

Análise de dados históricos para identificar relações, prever eventos ou observar tendências, embora a precisão dependa da qualidade dos dados arquivados.

03- Análise estatística

Para cumprir suas funções, engenheiros precisam entender como as características de produtos e processos variam, especialmente nas relações entre variáveis. Estudos geralmente são feitos com base em amostras e por isso é essencial dominar técnicas estatísticas para garantir que esses resultados possam ser generalizados.

Unidade de análise

É a unidade escolhida para observação. Pode variar conforme o contexto e o foco da análise, mas, em geral, representa o menor componente de software que será detalhadamente examinado.

População e amostra

A população é o conjunto total de itens do estudo (a população do estudo e a população alvo podem não ser as mesmas).

A amostra é um subconjunto representativo da população. A seleção de uma amostra tem base na sua representatividade.

Evento e distribuição

O evento representa uma situação que pode exigir uma resposta dentro de um sistema de software. São fundamentais em aplicações que reagem a mudanças no ambiente.

Já o intervalo e o padrão de variação de uma variável aleatória são dados por sua distribuição.



Correlação e regressão

Correlação

Mede a força da relação linear entre duas variáveis usando o coeficiente de correlação, que varia de -1 a +1. Um valor de +1 ou -1 indica uma relação perfeita, onde uma variável pode ser estimada com precisão a partir da outra. Valores positivos indicam que ambas as variáveis aumentam ou diminuem juntas, enquanto valores negativos indica, que uma aumenta à medida que a outra diminui.

Regressão

Além de medir a relação, busca modelar essa relação para fazer previsões. A força da relação é medida pelo coeficiente de determinação que varia de 0 a 1, onde 1 indica uma relação perfeita.



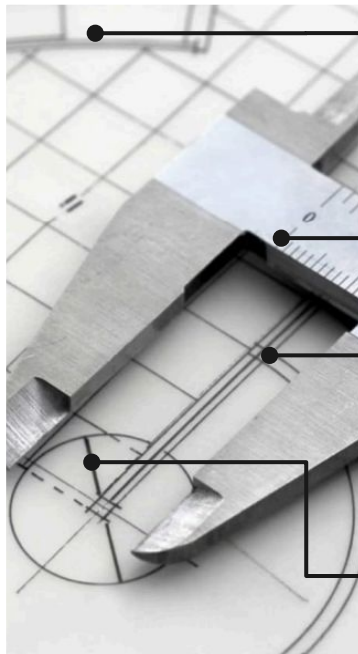
Medição

A medição é fundamental para obter dados quantitativos precisos e confiáveis em engenharia. Os métodos de medição variam de acordo com a natureza da grandeza a ser medida, mas em geral envolvem a comparação com um padrão de referência.

Na engenharia, a medição é utilizada para determinar o tamanho, a forma, a massa, a temperatura, a pressão, o fluxo e outras propriedades de objetos, materiais e sistemas. Ela é essencial para o projeto, a fabricação, o controle de qualidade e a operação de produtos e processos.

Definir medidas operacionais é essencial em engenharia, pois até medições aparentemente simples, como a altura, podem variar devido a fatores específicos. Engenheiros precisam de medidas bem definidas para garantir a precisão e consistência dos resultados.

Medição



1- Identificar a variável

Definir claramente a grandeza a ser medida.

2- Selecionar o método adequado

Escolher o método de medição mais preciso e adequado para a variável.

3- Realizar a medição

Utilizar o método de medição de forma correta e precisa para obter dados confiáveis.

4- Analisar os resultados

Verificar a precisão e confiabilidade dos dados coletados e realizar análises estatísticas, se necessário.

Projeto de engenharia

Definição do problema

Identificar e definir claramente o problema a ser resolvido pelo projeto.

Geração de ideias

Criar e explorar várias soluções potenciais para o problema.

Análise e seleção

Avaliar as soluções candidatas com base em critérios técnicos, econômicos e ambientais.

Implementação

Desenvolver o projeto detalhada, criar protótipos e construir a solução final

Teste e avaliação

Verificar o desempenho do projeto e realizar ajustes, se necessário, para otimizar a solução.



Modelagem, Simulação e Prototipagem

Modelagem

Criação de representações matemáticas ou computacionais de sistemas reais para análise e simulação.

Simulação

Executar modelos computacionais para testar o comportamento de sistemas sob diferentes cenários e condições.

Prototipagem

Criar versões físicas ou digitais de um projeto para testes e validação antes da produção em massa.



Normas

Padrões de segurança

Normas que garantem a segurança de pessoas e bens durante a operação de equipamentos e sistemas.
Protegem a integridade e privacidade dos dados.

Padrões de desempenho

Normas que especificam os requisitos de desempenho para equipamentos e sistemas, como eficiência energética e eficácia.
Evitam o uso excessivo de recursos e garantem que o software funcione da melhor maneira.

Padrões de qualidade

Normas que definem requisitos para o desempenho, confiabilidade e durabilidade dos produtos e serviços.
Garantem que o software seja funcional, seguro, confiável e fácil de usar



Application of Software Engineering Fundamentals: A Hands on Experience

A engenharia é construída sobre fundamentos sólidos, como cálculos precisos, análises rigorosas e a aplicação de normas e padrões.

O artigo discute como erros podem ocorrer no desenvolvimento de um software, demonstrando que engenharia exige não apenas conhecimento técnico, mas também planejamento cuidadoso, testes rigorosos e uma constante atenção aos detalhes para minimizar os riscos e garantir a excelência na execução.

Análise

1- Requisitos

No projeto, ocorreram falhas em relação ao atraso de entregas de requisitos.

Principais razões pelo atraso na conclusão de especificações de requisitos de software:

- Falta de informações nos documentos recebidos do cliente.
- Má interpretação da complexidade do projeto.
- Restrições de tempo.
- Falta de formação técnica em algumas áreas do projeto.

2- Processo de desenvolvimento: Alocação de esforço no projeto

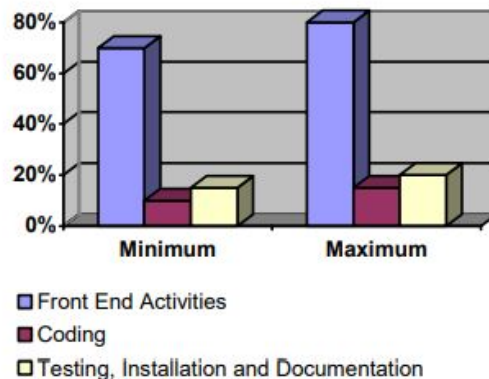


Fig. 1. Effort Allocation in the Project

Análise

Cronograma e acompanhamento do projeto

Devido a mudanças constantes na especificação de requisitos e riscos que não foram planejados e considerados quando se iniciou o projeto, ocorreram atrasos em alguns prazos de entrega.

Conclusão

Na prática, ao gerenciar projetos de desenvolvimento de software, a consciência de complexidade ao especificar requisitos, preparação de um documento bem escrito e a distribuição e agendamento de tarefas podem ter um impacto positivo sobre a produtividade do projeto.

TABLE 1
SOME PROJECT TASKS and THEIR SCHEDULE

Tasks	Planned Duration	Actual Duration
SRS Document	3 days	19 days
UML Use Case and Class Diagrams	2 days	On time
UML Sequence Diagrams	2 days	3 days
Information Gathering and Database Design	10 days	12 days
Coding	6 days	On time